Automatic Inference of Elevation and Drainage Models from a Satellite Image

ROBERT M. HARALICK, FELLOW, IEEE, JAMES B. CAMPBELL, AND SHYUAN WANG

Invited Paper

Oblique illumination of irregular topography generates a pattern of highlighting and shadow, as the solar beam directly illuminates those slopes that face the sun while those on opposite sides of ridgelines are shadowed. On remotely sensed images these patterns appear as alternating dark and bright regions that reveal approximate positions of ridges and valleys. Knowledge of scene-specific variables (such as sun angle and elevation), general knowledge of geomorphology, atmospheric scattering, and spectral characteristics of landscapes permits reconstruction of the topography from its manifestation on the image.

Raw image data record combined effects of topography, atmosphere, and diverse spectral reflections of surface materials. Our interpretation procedure isolates these several effects. From varied brightnesses caused by direct and indirect illumination, positions of ridges and valleys can be approximated. From variations in material reflectance, large rivers (channels with large areas of open water) can be detected. Finally, relative elevations can be estimated from analysis of drainage and ridge patterns using a strategy of "elevation growing" that assigns increasing elevation values to pixels as they are positioned at greater distances from rivers or other valley pixels already assigned elevations.

From the estimated topographic elevations, it is possible to derive a network of drainage channels. Each stream segment in this network is labeled with information pertaining to its length, junction with other segments, direction of flow, and other properties. We then examine this network to detect logical inconsistencies in the labeling of stream segments, then apply a procedure that identifies the optimal labeling to yield the smallest error within the network.

INTRODUCTION

Information describing the drainage pattern of a region can be derived by interpretation of multispectral data, and application of relational reasoning to resolve inconsistencies in initial approximations of the network. The result is an automated interpretation of a drainage network using a minimum of scene-specific prior information, and a minimum of interaction with the analyst.

Our method permits inference of elevation and drainage from a Landsat scene of mountainous terrain. To a first-order effect, the cause of the intensity value at any pixel is due to topographic orientation of the terrain. Topographic slopes facing the solar beam, are directly illuminated; there are additional effects due to the angle at which the sun illuminates this ground patch, and the reflectance of the surface material on the ground patch. To make sense of spatial patterns of image brightness values requires separation of these several effects. For this purpose, we use a clustering technique on ratio images to define homogeneous reflectance classes and then to perform a subclustering on these classes to separate directly lit from indirectly lit pixels. This subclustering creates a shadow image. Then we modify the Eliason, Soderblom, and Chavez [5] technique to create two images from the one Landsat image [3]. The first image is a "reflectance" image; the second is a topographic modulation image portraying information related to surface slope and sun illumination.

From this estimate of the topography, it is possible to trace valleys, and therefore to derive a network of the drainage system. Because the organization of the drainage network follows a consistent logic it is possible to detect errors in the original estimation of elevations and drainage channels. Therefore, the final step in our interpretation is an examination of the lengths of stream segments, and the angles at which they join, to estimate the direction of flow within the drainage network. We complete this process by assigning labels to all stream segments, to identify the positions of stream segments within the drainage hierarchy. If errors are present, then the assigned labels violate the known logic of a drainage system. By using a procedure known as "forward checking" it is possible to examine the impact of a specific error upon the entire system, and therefore to select the set of labels that produce minimum error throughout the network.

Knowledge of solar azimuth and the shadow image constitute sufficient information for identification of ridges and valleys. With the valleys identified, each valley pixel may be assigned a relative elevation value that increases as distance from the mouth of the valley increases. Ridges must be assigned elevations higher than their neighboring valleys

Manuscript received October 1, 1984; revised December 26, 1984.

R. M. Haralick is with Machine Vision International, Ann Arbor, MI 48104, USA.

J. B. Campbell is with the Department of Geography, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA.

S. Wang is with the Department of Computer and Information Science, George Mason University, Fairfax, VA 22030, USA.

0018-9219/85/0600-1040$01.00 ©1985 IEEE

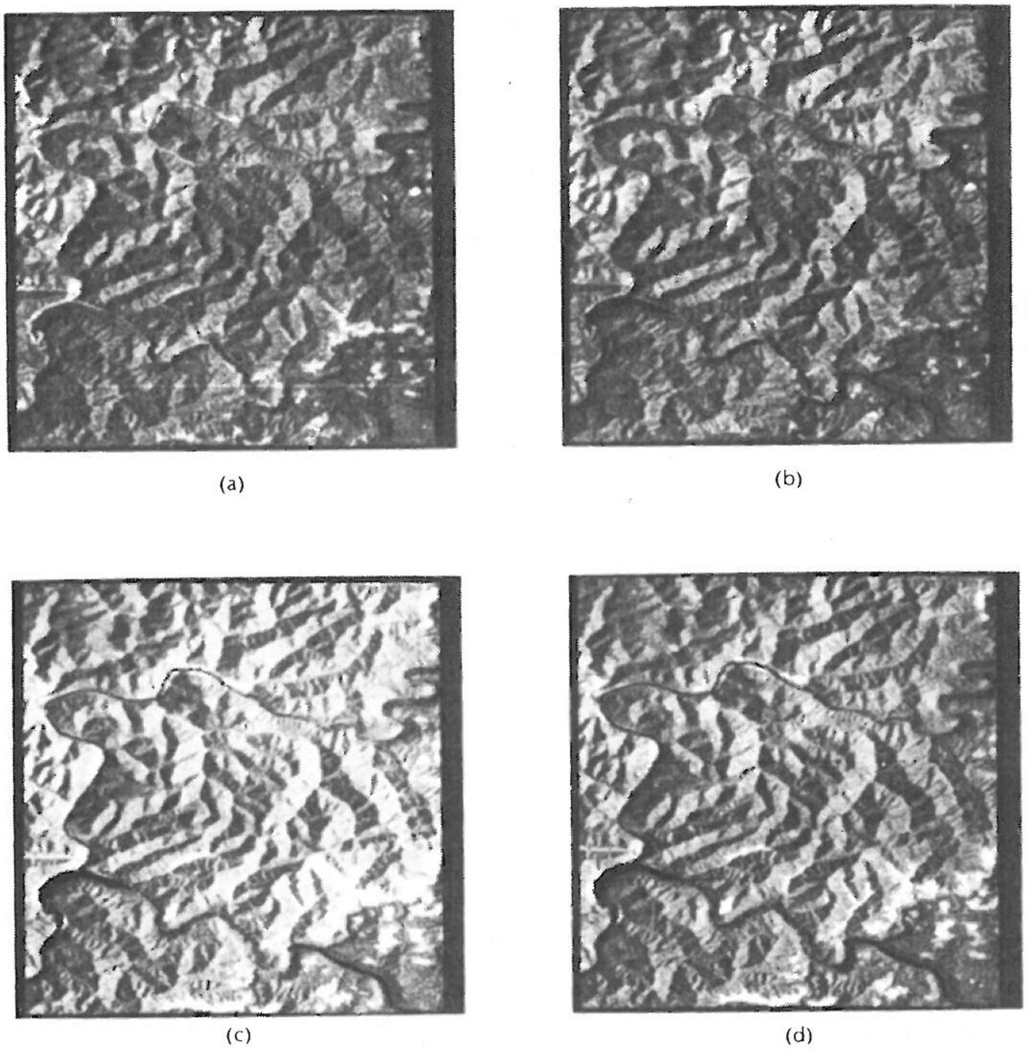

(a)
(b)
(c)
(d)

Fig. 1. Original Landsat subscene. (a) Band 1. (b) Band 2. (c) Band 3. (d) Band 4.

such that elevations of crest decrease along the path from a peak to the saddle point where the ridge crosses a valley. In order to do this, local slopes must be known. Some estimated local slopes are assigned initially to generate the first version of elevation model. Then the topographic modulation image can be used to calculate more accurate local slopes to improve the elevation values [22]..

Study Area

This research examines an area in southeastern West Virginia (Fig. 1). This region is a portion of the Appalachian Plateau physiographic province, within the "unglaciated Allegheny plateau" described by Thornbury [18]. In general, this region is a thoroughly dissected plateau-like surface. It receives about 1 m of precipitation each year and, as depicted on topographic maps, has a moderate drainage network density. Drainage is through tributaries of the New (Kanawha) River, which flows west into the Ohio River drainage system.

Overall drainage within this region consists of a relatively large sinuous channel (the Gauley River) superimposed over the finer texture of a dentritic pattern formed by smaller streams. A number of small rivers flow directly into the large channel. Thus the overall pattern is composed of a mixture of numerous very small stream segments, many with very steep gradients, a prominent major channel with a relatively low gradient, and relatively few stream segments of intermediate length and gradient.

Throughout the area, flood plains (when present) are narrow and tend to closely follow the course of the stream channel. Valleys are narrow, with steep sides; the Gauley River, for example, follows a valley that is typically 150 m deep but only 100 m wide. Uplands often consist only of ridge crests; although plateau-like upland regions are present, they are not continuous or extensive. The area is forested with a dense cover of deciduous trees (Kuchler's "mixed mesophytic forest" [11]). Cleared areas for agriculture (chiefly pasture) tend to follow the valleys of intermediate-sized streams. Settlements are small and dispersed, usually positioned in valleys.

This region appears on the Charleston, West Virginia/Ohio USGS 1:250000 quadrangle (NJ 17-5). Our investigations include areas in Nicholas County, W VA and neighboring counties. This area was imaged by the Landsat-1 MSS on April 13, 1976 (scene id: 5360-14502; path 18, row 34). For our study we have subset the original Landsat scene to produce a smaller image of 257 pixels on a side (Fig. 1). The date of the image reflects important qualities of the scene. First, at this date, the atmosphere was quite clear—there is no evidence of severe atmospheric (Mie) scattering or degradation of the data. Also, at this spring date most of the forested areas are without leaves, especially at higher elevations. Lower elevations have a cover of newly emerged leaves and grasses. Within a few weeks leaves will have emerged in vegetation throughout the entire region, but at this time in April, there is a sharp spectral contrast between the vegetation cover of the higher elevations (which is without foliage) and that of some of the valleys (which includes areas covered by dense vegetative cover). This contrast is especially sharp in bands 3 and 4.

The Problem of Mixed Information

Four kinds of information are mixed in Landsat imagery: surface reflectance, topography, diffuse light, and haze. Assuming the ground surface is flat, vegetated areas have high reflectance for some spectral regions and appear as bright areas to the Landsat sensor. On the other hand, areas of open water have low reflectance, so they appear as dark areas to the Landsat sensor. Topography manifests itself as a pattern of alternating directly illuminated and shadowed slopes caused by highlighting of slopes facing the solar beam, and shadowing of those facing away from the sun. However, graytones for image pixels corresponding to shadowed locations are not zero because of diffuse light caused by atmospheric scattering within shadows. Finally, when light is reflected from the ground back to the sensor, there is additive brightness due to scattering by atmospheric haze. The difficulty of interpreting Landsat scenes of mountainous areas is due to the mixing of topographic and reflectance data. To separate these individual components we need to consider an illumination model.

Separating the Information

The basic data model for a Lambertian surface illuminated by a point source is

$$G(x,y) = r(x,y)\,I\cos\theta(x,y) \tag{1}$$

where

G brightness value of a pixel within the image,
x, y pixel coordinates,
r surface reflectance,
I the illumination flux from the sun,
θ the angle between sun incidence direction and surface normal (Fig. 2).

Adding band number, diffuse light, and haze into this model, one has the general model for Landsat data as follows:

1) For directly illuminated pixels

$$G(x,y,b) = r(x,y,b)\,I(b)\cos\theta(x,y) + r(x,y,b)\,D(b) + H(b). \tag{2}$$

2) For shadowed pixels

$$G(x,y,b) = r(x,y,b)\,D(b) + H(b)$$

where

b the spectral band number,
D diffuse light,
H the effect of scattering due to atmospheric haze.

Because haze is an additive constant independent of pixel locations, we use the Switzer, Kowalik, and Lyon [17] technique for haze removal. The haze corrected image $G - H$ is defined as G'.

After haze is removed, it can be seen that resolution of the remaining components amounts to extracting diffuse light Df containing the information of $r(x,y,b)D(b)$, reflectance data R which contain the information of $r(x,y,b)I(b)$, and topographic modulation data Tp which contain the information of $\cos\theta(x,y)$.

1) For directly illuminated pixels

$$G'(x,y,b) = R(x,y,b)\,Tp(x,y) + Df(x,y,b). \tag{3}$$

2) For shadowed pixels

$$G'(x,y,b) = Df(x,y,b).$$

To unravel these two sources of brightness, it is first necessary to determine those pixels which are directly illuminated and those pixels which are in shadow. Once this is accomplished the unraveling can begin. For example, for the diffuse light image, shadowed pixels take their value as the dehazed data value. Directly illuminated pixels assume brightness equal to the average dehazed value taken over all shadowed pixels determined to have similar spectral properties (as described below).

To separate the shadow pixels from the directly lit pixels, we seek to transform the images in a way in which the only effect is reflectance. Then within groups of pixels with similar reflectance, we can separate the bright appearing ones from the dark appearing ones. This two-step technique is more accurate than a simple thresholding technique [21].

One way to transform the data so that the only remaining effect is reflectance is to take ratios of one band to another. The ratio image has been widely used by remote sensing researchers to subdue surface topographic effects [5], [20]. From (2), the ratio image of two bands with band numbers $b1$ and $b2$ for directly illuminated pixels after haze is removed is

$$\frac{G'(x,y,b1)}{G'(x,y,b2)} = \frac{r(x,y,b1)[I(b1)\cos\theta(x,y) + D(b1)]}{r(x,y,b2)[I(b2)\cos\theta(x,y) + D(b2)]}.$$

Spectral transmission functions for radiation passing through clouds are certainly different than those for a clear atmosphere. However, to a first approximation we assume these two functions to be proportional and therefore

$$I(b1) = aI(b2)$$
$$D(b1) = aD(b2)$$

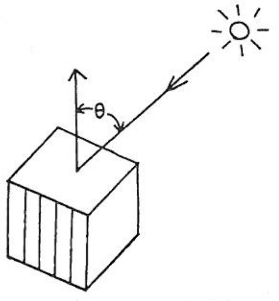


Fig. 2. θ is the angle between sun incidence and surface normal.

then

$$\frac{G'(x,y,b1)}{G'(x,y,b2)} = \frac{r(x,y,b1)\,a[I(b2)\cos\theta(x,y) + D(b2)]}{r(x,y,b2)[I(b2)\cos\theta(x,y) + D(b2)]}$$
$$= a\frac{r(x,y,b1)}{r(x,y,b2)}. \qquad (4)$$

Similarly, the ratio image for shadowed pixels is

$$\frac{G'(x,y,b1)}{G'(x,y,b2)} = a\frac{r(x,y,b1)}{r(x,y,b2)}. \qquad (5)$$

Thus whether shadowed or directly illuminated, the ratio image is independent of $\cos\theta$. Three independent ratio images taken from the 4-band imagery in Fig. 1 are shown in Fig. 3. It can be clearly seen that the effects of shadows have been removed.

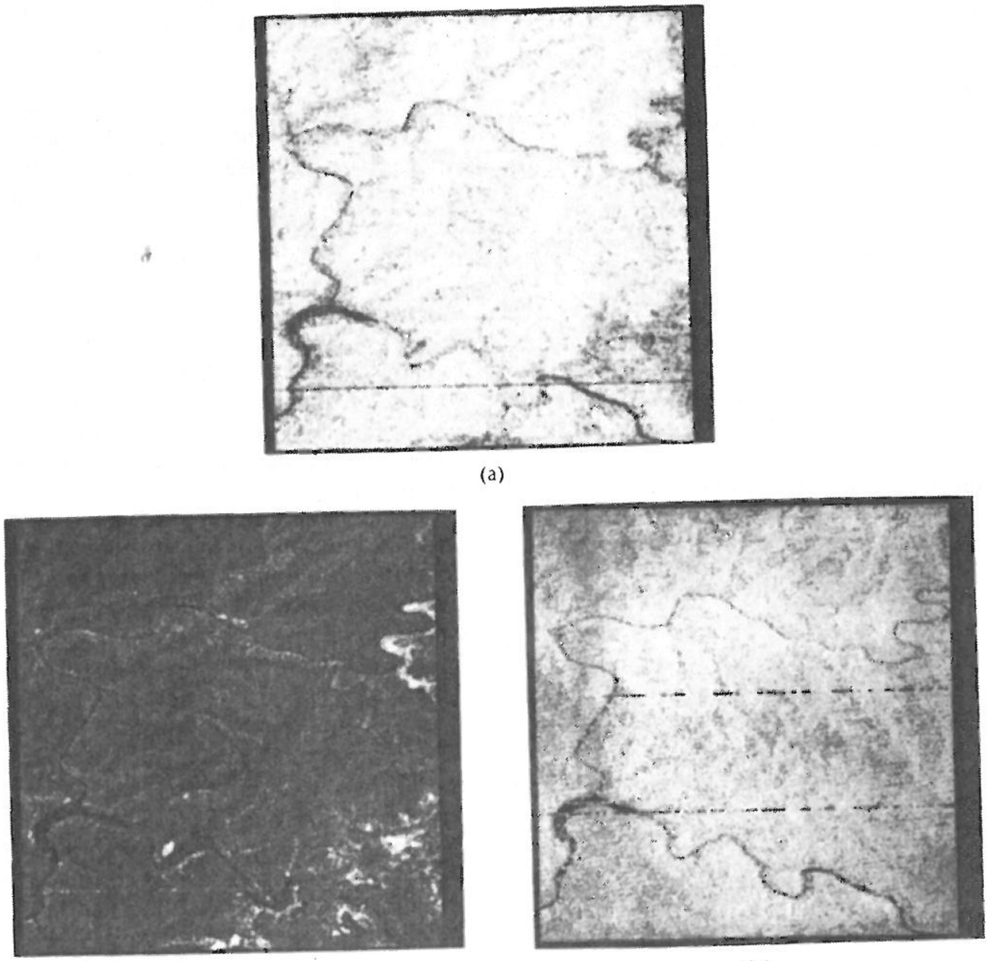
Clustering

We assume that materials which are the same have the same spectral response. Materials which are different have a different hue. Because the three ratio images depend upon material reflectance only, regions of the same material reflectance can be identified by grouping together pixels of similar spectral characteristics. Because we desire to conduct the analysis using a minimum of prior information, unsupervised classification is favored over the supervised approach, which would require detailed knowledge of the number, identity, and characteristics of groups. Unsupervised classification permits identification of the natural structure of the image with a minimum of prior information.

In the noisy ratio images of Fig. 3, there are three major clusters: open water, vegetated regions, and nonvegetated areas. The size of the water area is much smaller than that of the other two. For this reason, the mode approach which uses a fixed threshold to get cluster centers does not work. AMOEBA [2] works better, but fails to obtain unbroken river segments. It was found that ISODATA [4], [12] modified in such a way that class sizes are also considered works best for our purpose (Fig. 4).

Once the material clusters are defined on the basis of the ratio images, one can find directly illuminated and shadowed pixels and define a binary shadow image (Fig. 5). To do this, we collect together all dehazed 4-band pixel values belonging to a single material cluster and subcluster these 4-tuples into dark and bright subcluster classes. The next few paragraphs describe this in detail.

If one overlays the material cluster image *Mc* over any band of the dehazed image, one can see, within each material cluster, some pixels are bright and the others are dark. These differences are due to variations in topography; the bright pixels are directly illuminated pixels, and the dark pixels are in shadow. To separate the shadow pixels from



(a)

(b)

(c)

Fig. 3. Ratio images. (a) Band 2/band 1. (b) Band 3/band 2. (c) Band 4/band 3.

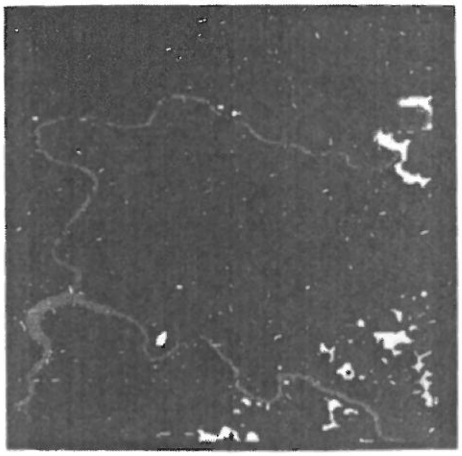


Fig. 4. Material cluster image.

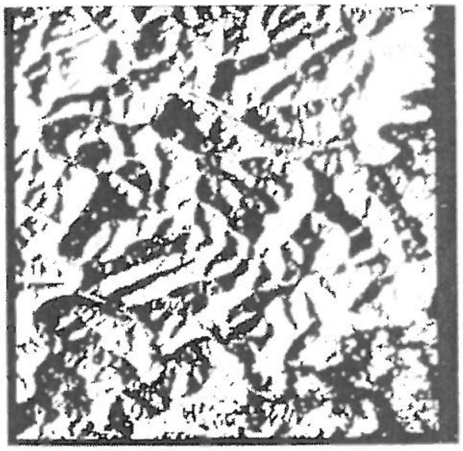


Fig. 5. Binary shadow image.

the directly lit pixels, for each material cluster $c1$, one performs a subclustering on dehazed pixel values in the set

$$\{ G'(x,y,b) \mid Mc(x,y) = c1 \}$$

which is the set of all dehazed values for pixels whose material cluster index is $c1$. This subclustering on cluster $c1$ separates the directly illuminated pixels $C0(c1)$ from the indirectly illuminated pixels $C1(c1)$

$$C0(c1) = \{(x,y) \mid (x,y) \text{ is directly illuminated on the basis of the subclustering}\}$$

$$C1(c1) = \{(x,y) \mid (x,y) \text{ is indirectly illuminated on the basis of the subclustering}\}.$$

The subclustering that produces $C0$ and $C1$ uses only the basic ISODATA program. In this case, the initial class mean for $C1$ includes all the minimum graytones for four bands, and the initial mean for $C0$ includes all the maximum graytones for four bands.

A shadow image Sw can be defined as

$$\begin{aligned} &Sw: XxY \rightarrow \{0,\ 1\} \\ &Sw: (x,y) = \begin{cases} 0, & \text{if } (x,y) \in C0(Mc(x,y)) \\ 1, & \text{if } (x,y) \in C1(Mc(x,y)). \end{cases} \end{aligned} \tag{6}$$

The shadow image for Fig. 1 is shown as Fig. 5. The correspondence between this image and the topographic map is quite good. Now that directly lit and shadowed pixels have been identified, it is possible to use the dehazed image to find the diffuse light image Df, the reflectance image R, and the topographic modulation image Tp.

Diffuse Light Data, Reflectance Data, and Topographic Modulation Data

From (3), shadowed pixels contain only the information of diffuse light, but directly illuminated pixels contain the information of both diffuse light and direct sun illumination. By the method of clustering, each pixel in the image belongs to a material cluster. Each material cluster has a bright and a dark subcluster. For pixels in the dark subcluster, one can simply use their dehazed values as their values in the diffuse light image Df. On the other hand, for pixels in the bright subcluster, one can define their value in the diffuse light image to be the average value of all the pixels from the dark subcluster associated with the material cluster to which these pixels belong. With this definition, the diffuse light image Df is as follows:

1) For directly illuminated pixels

$$Df(x,y,b) = \sum_{\substack{(u,v) \in C1(c1) \\ c1 = Mc(x,y)}} \frac{G'(u,v,b)}{\#C1(c1)}$$

where, for a set S, $\#S$ means the size of this set.

2) For shadowed pixels

$$Df(x,y,b) = G'(x,y,b).$$

If there are no variations in reflectance for pixels from the same material, we have

Assumption 1: $r(x,y,b)$ is a constant $r'(c1,b)$ for all pixels (x,y) in $C(c1)$, where $c1 = Mc(x,y)$.

Under this assumption, for directly illuminated pixels

$$\begin{aligned} Df(x,y,b) &= r'(c1,b)\,D(b) \sum_{(u,v) \in C1(c1)} \frac{1}{\#C1(c1)} \\ &= r'(c1,b)\,D(b). \end{aligned} \tag{7}$$

From (2), (7), and Assumption 1, $G' - Df$ is

1) For directly illuminated pixels

$$\begin{aligned} &G'(x,y,b) - Df(x,y,b) \\ &\quad = r(x,y,b)\,I(b)\cos\theta(x,y) + r(x,y,b)\,D(b) \\ &\qquad - r'(c1,b)\,D(b) \qquad (c1 = Mc(x,y)) \\ &\quad = r(x,y,b)\,I(b)\cos\theta(x,y). \end{aligned}$$

2) For shadowed pixels,

$$G'(x,y,b) - Df(x,y,b) = 0. \tag{8}$$

The Df image for Fig. 1 is shown in Fig. 6; these show the diffuse light components original image, as isolated from the other effects that contribute to observed brightness in Fig. 1.

An initial or raw estimated reflectance image R' can be calculated by assigning each pixel's value to be the average $G' - Df$ value of all the pixels from the bright subcluster associated with the material cluster to which the pixel belongs.

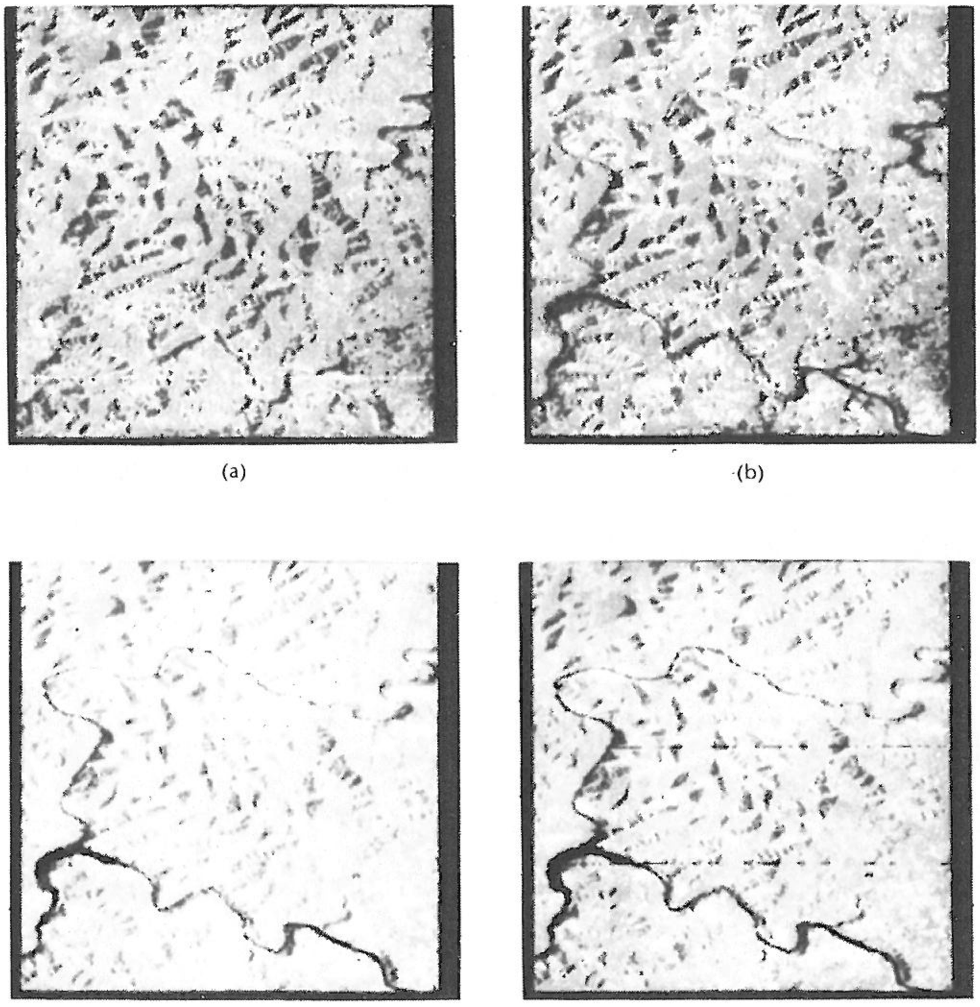

(a) (b) (c) (d)

Fig. 6. Diffuse light images that show the brightness of diffuse light in each band, as separated from other components. (a) Band 1. (b) Band 2. (c) Band 3. (d) Band 4.

$$R'(x,y,b) = \sum_{\substack{(u,v)\in C0(c1)\\ c1=Mc(x,y)}} \frac{G'(u,v,b)-Df(u,v,b)}{\#C0(c1)}$$

$$= \sum_{(u,v)\in C0(c1)} \frac{r(u,v,b)\,I(b)\cos\theta(u,v)}{\#C0(c1)}$$

$$= r'(c1,b)\,I(b) \sum_{(u,v)\in C0(c1)} \frac{\cos\theta(u,v)}{\#C0(c1)}.$$

Define

$$Xc(c1) = \sum_{(u,v)\in C0(c1)} \frac{\cos\theta(u,v)}{\#C0(c1)}$$

where $Xc(c1)$ is the spatial average of $\cos\theta$ for pixels in the bright subcluster. Then

$$R'(x,y,b) = r'(c1,b)\,I(b)\,Xc(c1). \qquad (9)$$

We use the raw estimated reflectance images to produce raw estimates of the topographic modulation image. By (3), (8), and (9), the raw estimated topographic modulation image Tp' for band b is as follows:

1) For directly illuminated pixels

$$Tp'(x,y,b) = \frac{G'(x,y,b)-Df(x,y,b)}{R'(x,y,b)}$$

$$= \frac{r(x,y,b)\,I(b)\cos\theta(x,y)}{r'(c1,b)\,I(b)\,Xc(c1)} \qquad (c1 = Mc(x,y))$$

$$= \frac{\cos\theta(x,y)}{Xc(c1)}.$$

2) For shadowed pixels

$$Tp'(x,y,b) = 0. \qquad (10)$$

We use the following assumption to convert the raw topographic images to our final estimated topographic modulation image.

Assumption 2: $Xc(c1)$ is the same for all material clusters, $1 \leqslant c1 \leqslant Nc$.

With this assumption, the principal component image generated from Tp' corresponding to the largest eigenvalue becomes our final single-band estimated Tp image.

At this point, we have a four-band image Tp each band of which is an estimate of $\cos\theta\ (x,y)$. The first principal component of this four-band image provides an estimate which is close to the minimum variance estimate of Tp.

1) For directly illuminated pixels

$$Tp(x,y) = k\cos\theta(x,y), \qquad \text{for a constant } k. \tag{11}$$

2) For shadowed pixels

$$Tp(x,y) = 0.$$

Having the topographic modulation image, the final estimated reflectance image is easily computed using (3).

1) For directly illuminated pixels

$$\begin{aligned} R(x,y,b) &= \frac{G'(x,y,b) - Df(x,y,b)}{Tp(x,y)} \\ &= \frac{r(x,y,b)\,I(b)\cos\theta(x,y)}{k\cos\theta(x,y)} \\ &= \frac{r(x,y,b)\,I(b)}{k}. \end{aligned} \tag{12}$$

2) For shadowed pixels

$$\begin{aligned} R(x,y,b) &= \sum_{\substack{(u,v)\in C0(c1)\\ c1=Mc(x,y)}} - \frac{R(u,v,b)}{\#C0(c1)} \\ &= \frac{I(b)}{k}\sum_{(u,v)\in C0(c1)} \frac{r(u,v,b)}{\#C0(c1)}. \end{aligned}$$

The Tp image is shown in Fig. 7, and the R image is shown in Fig. 8.

Elevation Estimation

Identification of Ridges and Valleys

In the last section, the problem of confounded data is handled in such a way that the material information is contained in the reflectance image and the diffuse light image. The topographic information is contained in the

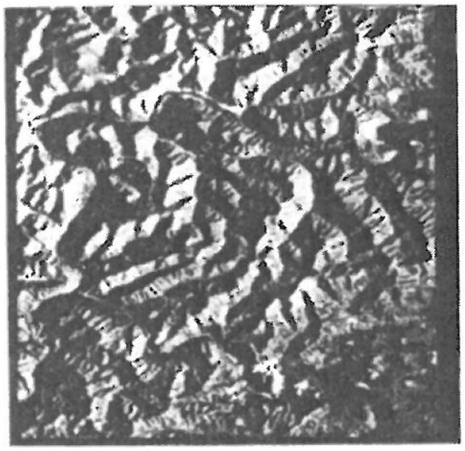
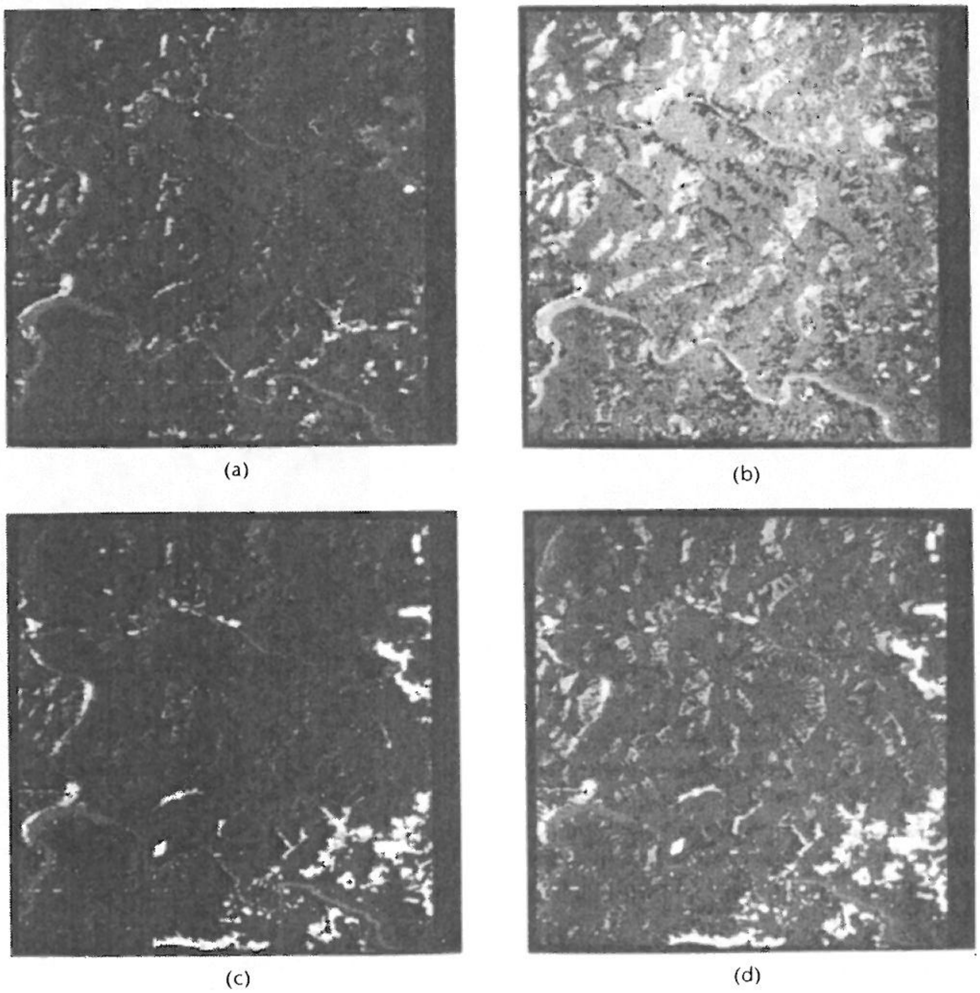


Fig. 7. Topographic modulation image.



(a)

(b)

(c)

(d)

Fig. 8. Reflectance image. (a) Band 1. (b) Band 2. (c) Band 3. (d) Band 4.

shadow image and the topographic modulation image. In this section, we will show how to detect ridge segments and valley segments from the shadow image. In the next section, we will perform an elevation growing to obtain initial raw estimates of elevations for all pixels on the basis of these ridge and valley segments.

Sides of hillsides facing the sun must be directly lit. Sides of hillsides facing away from the sun must be indirectly lit. A directly to indirectly lit transition in a direction moving away from the sun is a ridge. An indirectly to directly lit transition in a direction moving away from the sun is a valley. Thus valleys and ridges exist on the borders between shadowed and directly lit areas. To find these areas we use the binary shadow image. First, a connected components operation determines regions on the shadow image. Then small, noisy regions are eliminated.

Next, the perimeters of these bright and shadowed regions are segmented into border segments according to their left regions, right regions, and orientations. A border segment is a maximally long sequence of connected pixels which are on the border between two given regions. Because the detection of ridges and valleys is highly orientation-dependent and the sun illumination comes from the east in Fig. 1, each border segment is further broken into several pieces according to orientation; all the east-west parts are separated from the north-south parts. The result is shown in Fig. 9. (In this figure, the individual symbols that designate separate border segments are not visible.)

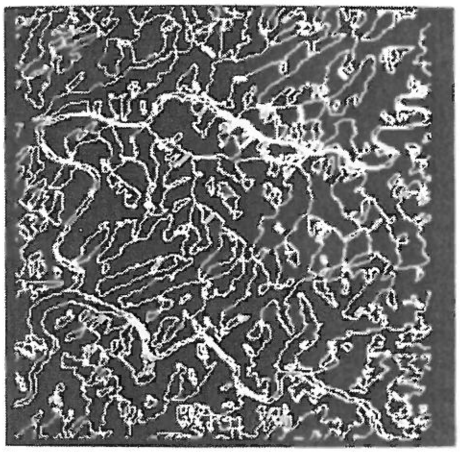


Fig. 9. Border segments.

As the sun illumination is from the east in our Landsat image, those border segments which are valley segments or ridge segments can be identified according to the brightness of the regions adjacent on the left and on the right. Because most of the trees in this area in April are unfoliated, the strongest region boundaries are shadow boundaries rather than reflectance boundaries, and the strongest boundaries are those at the extremes of steep slopes oriented normal to the sun direction. Because the sun illumination is predominantly east-west, a boundary that is dark on the left and bright on the right will correspond to a ridge, and the reverse will correspond to a valley.

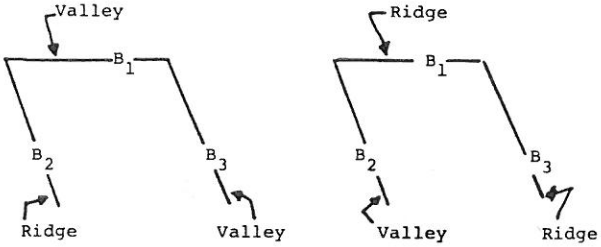
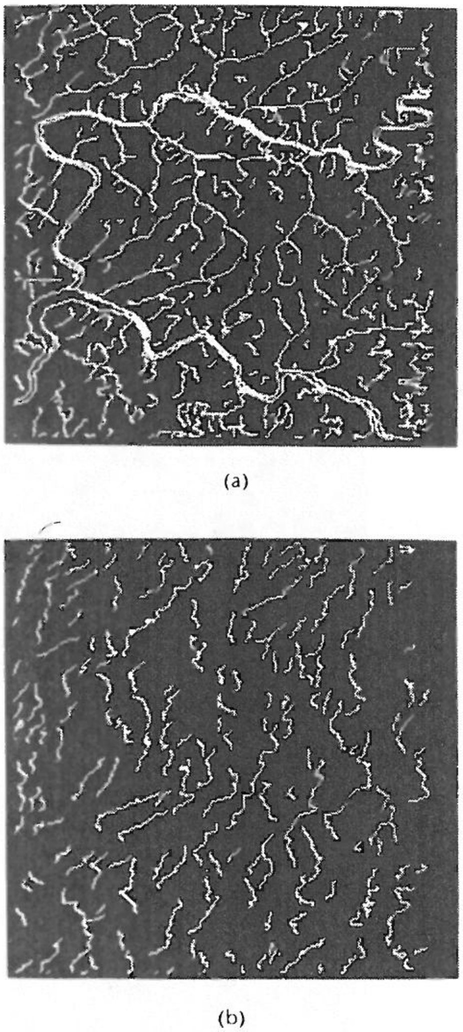


Fig. 10. Classifying east-west border segments. "B_1" represents east-west boundary. "B_2" designates a north-south border joining B_1 at the west (left) end. "B_3" designates a north-south border joining B_1 at the east (right) end. The angles of junctions of B_1 with B_2 and B_3 permit labeling of B_1 as explained in the text. For the situation on the left B_1 is assigned the valley label from B_3; on the right B_1 is assigned the ridge label from B_3.

East-west region boundaries are classified according to the labeling of neighboring north-south boundaries as well as their orientation relative to the east-west boundaries. As shown in Fig. 10, each east-west boundary B_1 has a left intersecting north-south boundary B_2 and a right intersecting north-south boundary B_3. If the angle between B_1 and B_2 is smaller than the angle between B_1 to B_3, then we assign the labeling of boundary B_2 to B_1; otherwise, we assign the labeling of boundary B_3 to B_1. The results of ridge-valley finding are shown in Fig. 11.



(a)

(b)

Fig. 11. (a) Valley map consisting of border segments identified as valleys. (b) Ridge map consisting of border segments identified as valleys. The valley and ridge maps are both derived from the set of all border segments, as described in the text and Fig. 10.

Elevation Growing

The detection of the ridge and valley segments as discussed in the last section only assigns a ridge or valley label, but does not assign relative elevations. This section describes how to estimate relative elevations. First, a model called elevation growing is used to assign initial estimated elevations for all ridge and valley pixels. Next, interpolation assigns elevations for non-ridge and non-valley pixels.

The cross sections of valleys are V shaped, and the cross sections of ridges are A shaped. If one looks at topographic maps, the elevation contours of valleys such as those shown in Fig. 12 can be frequently found. Thus if one draws a line *ab* perpendicular to the valley axis, the elevations are increasing from point *o* to point *a*, and also from point *o* to point *b*. If the end point of a valley segment of smaller order is encountered during the growing, it is deduced that this end point is the lower end of this smaller valley segment. However, if a ridge point is encountered during the process, the increasing has to stop because the elevation starts to decrease. Based on this knowledge, an "elevation growing" model can be created.

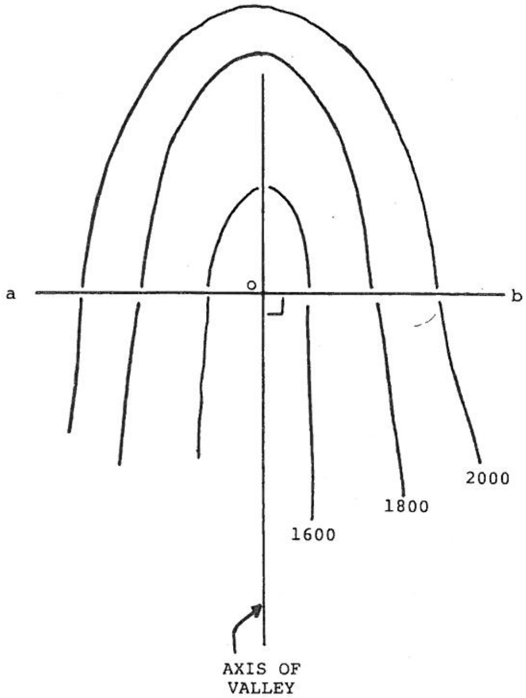


Fig. 12. Elevation pattern of valleys and its relation to elevation growing.

Three different local slopes are assigned to three classes of non-ridge and non-valley pixels: pixels which are close to ridges, pixels which are close to valleys, and other pixels. A large local slope 0.4 m is assigned to pixels within 5 pixel distances to ridges so that one has steep hillsides; a small local slope 0.02 m is assigned to the four neighbors of valley pixels so that one has a wider valley bottom, and a medium local slope 0.1 m is assigned to the remaining pixels.

We have estimated these values from our knowledge of local topographic slopes. In instances where such knowledge is not available in detail from field observations, reasonable values can sometimes be estimated from general information concerning regional geology and geomorphology. Of course, the more detailed, and the more accurate these estimates are, the better the resulting elevation model.

Using estimates of local slopes and assuming the elevations of visible rivers, detected by the Alfoldi and Munday technique [1], are lowest in a small area, the elevation-growing algorithm can be defined as follows:

1) Trace the border segments of visible rivers and give all the pixels elevation E_0. E_0 can be an arbitrary constant or a datum read from the map.

2) Repeat until all the pixels are elevation labeled.

a) Growing:

If a pixel *p* has elevation $\mathrm{El}(p)$, give its unassigned neighbors elevations $\mathrm{El}(p) + \Delta_{(p)}$ ($\Delta_{(p)}$ is the assigned local slope) unless:

i) an image boundary is encountered

ii) a ridge is encountered.

b) Including new valley segments:

If any elevation-unlabeled valley segment is touched by an elevation-labeled pixel *pe* resulting from the growing, assign elevations to all the pixels of this segment. The end touching *pe* will have the same elevation as *pe*. Then starting from this end, trace the whole segment and give every pixel linearly increasing elevation with some constant slope.

3) For a ridge pixel, take the maximum elevation value from its four neighbors' as its elevation.

Because realistic shape of the hillsides from valleys to ridges were not taken into account in the raw elevation growing, only the relative elevations of the ridges and valleys are held to be accurate. Haralick *et al.* [10] describe a few interpolation procedures which permit more realistic elevation assignment to non-valley and non-ridge pixels. The interpolation makes all non-ridge or -valley pixels be the recursive average of their north, south, east, and west neighbors and makes ridge or valley pixels keep the values produced by the elevation growing model (Fig. 13).

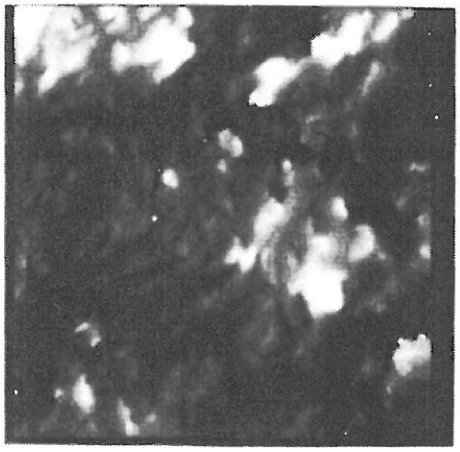


Fig. 13. Elevation model.

Refinement of the Network by Relational Reasoning

The valley segments within our elevation model form an approximation of the drainage network, as all valleys are either occupied by perennial streams, or form pathways for movement of moisture into these streams. Yet our initial estimate of the drainage network is imperfect.

Several factors combine to produce errors. In some areas, ridges are oriented parallel to the solar beam, and therefore do not cast shadows that permit them to be detected by our interpretation procedure. In other regions, effects of coarse spatial resolution, or noisy artifacts of our procedure produce gaps in the ridge and stream segments that can produce incorrect links when isolated segments are later joined to form a network of region boundaries. Therefore, although our elevation model is usually quite effective in identifying major topographic features, even a few relatively minor errors at critical points (at divides, or junctions) can cause inconsistencies in the drainage network.

Fortunately, many of these problems can be detected and resolved by applying rules that apply to the organization of a stream network. In most terrains, stream segments within a drainage system are organized in a logical, systematic manner. Flow is in a single direction downhill. Small streams feed into larger streams. Junctions between segments follow certain relationships.

From this point forward we consider our stream network simply as a set of line segments that must be spatially organized according to a specific logic. If we discover a spatial arrangement that violates this logic, we know that an error is present somewhere in the network. Then the network can be repaired by changing it in a manner that produces the best correspondence of the revised network to the logic of the overall system. The following sections describe our procedure for implementing relational reasoning.

Valley Segments

In our problem, we are interested in assigning labels of {upstream, downstream} to the visible stream segments by looking at the constraints at junctions. For example, as shown in Fig. 14, when a smaller stream s_2 flows into a larger stream which is composed of two segments s_1 and s_3 because of this intersection, very often the angle between s_2 and s_1 is less than 90°. Some general rules governing flow directions at junctions are given below.

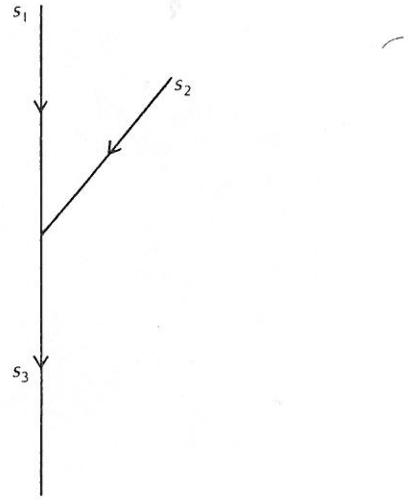


Fig. 14. Labeling of stream junctions.

Constraints at Junctions

It is believed that when several stream segments join at a junction there are constraints based on orientation and length patterns. A set of general rules about flow directions at junctions is given in Table 1, which is derived from a common-sense interpretation of lengths of segments and the angles at which they meet. These rules can be refined, or elaborated, if more information is known about sizes of channels [15]; $A(s_1, s_2)$ indicates the angle between segments s_1 and s_2 and $L(s)$ indicates the length of a segment s.

Table 1 Rules of Flow Directions at Junctions

Pattern Number	$A(s_1, s_3)$	$A(s_1, s_2)$	$A(s_2, s_3)$	Upstream Segment	Downstream Segment
	$= 180$	$= 90$	$= 90$		
1	$L(s_3) \geqslant \max(L(s_1), L(s_2))$			s_1 and s_2	s_3
2	$L(s_2) \leqslant \min(L(s_1), L(s_3))$			s_2	s_1 or s_3
3	$L(s_3) \leqslant \min(L(s_1), L(s_2))$			s_2 and s_3	s_1
4	$L(s_1) = L(s_2) = L(s_3)$			s_2	s_1 or s_3
5	$= 180$	< 90	> 90	s_1 and s_2	s_3
6	$= 180$	> 90	< 90	s_2 and s_3	s_1
7	< 180	$\geqslant 90$	$\geqslant 90$	s_1 and s_3	s_1
	> 180	$\geqslant 90$	$\leqslant 90$		
8	$L(s_1) = L(s_2) = L(s_3)$			s_2 and s_3	s_1
9	$L(s_2) \leqslant \min(L(s_1), L(s_3))$			?	
10	$L(s_1) \leqslant \min(L(s_2), L(s_3))$			s_2 and s_3	s_1
11	$L(s_3) \leqslant \min(L(s_1), L(s_2))$			?	
	$= 120$	$= 120$	$= 120$		
12	$L(s_1) = L(s_2) = L(s_3)$			?	
13	$L(s_2) \leqslant \min(L(s_1), L(s_3))$			s_1 and s_3	s_2
14	$L(s_1) \leqslant \min(L(s_2), L(s_3))$			s_1 and s_3	s_2
15	$L(s_3) \leqslant \min(L(s_1), L(s_2))$			s_1 and s_2	s_3

We are interested in two kinds of junctions. Junctions of the first kind are vertices at which three stream segments meet. The set of such junctions is called J_3. Junctions of the second kind are vertices at which two stream segments and one valley segment meet. The set of such junctions is called J_2. Here we define J as the union of sets J_3 and J_2 (i.e., $J = J_3 \cup J_2$). Also we call S the set of all stream segments and V the set of all valley segments.

Let $J = J_3 \cup J_2$, and let X be the set of junction patterns in Table 1, and $L = \{\text{upstream, downstream}\}$. Then one can define $a\colon J \rightarrow X$ as the function that assigns junction patterns to junctions. As an example, for the junction j in Fig. 14, $a(j) = 5$ because $A(s_1, s_3) = 180$, $A(s_1, s_2) < 90$, and $A(s_2, s_3) > 90$. For each pattern in Table 1, the flow directions of segments s_1, s_2, s_3 can be put in a triple which is an element in $L \times L \times L$. Thus the mapping from pattern numbers to flow directions can be defined as $b\colon X \rightarrow L \times L \times L$. For the junction j in Fig. 14, $b(a(j)) =$ (upstream, upstream, downstream).

If three stream segments meet at a junction, two constraint relations can be formally stated on the basis of Table 1 as follows. One is concerned with all triples of stream segments that constrain each other because they meet at this junction; the other is concerned with all triples of segment–label pairs where the stream segments meet in a junction and the labels are possible for that type of junction. For each x in X, we can define T_x and R_x as follows:

$$T_x = \{\{s_1, s_2, s_3\} | s_1, s_2, s_3 \in S, \text{ and } s_1, s_2, s_3 \text{ meet at a junction } j \text{ of type } x, j \in J_3\}$$

$$R_x = \{(s_1, \ell_1), (s_2, \ell_2), (s_3, \ell_3) | \{s_1, s_2, s_3\} \in T_x, \text{ and } (\ell_1, \ell_2, \ell_3) \in b(x)\}.$$

If two stream segments and one valley segment meet in a junction, two similar constraint relations can be stated as follows. For each x in X

$$T_x' = \{\{s_1, s_2\} | s_1, s_2 \in S \text{ and there exists } s_3 \in V \text{ such that } s_1, s_2, s_3 \text{ meet at a junction } j \text{ of type } x, j \in J_2\}$$

$$R_x' = \{(s_1, \ell_1), (s_2, \ell_2) | \{s_1, s_2\} \in T_x', \text{ and } (\ell_1, \ell_2, \ell_3) \in b(x)\}.$$

Now let

$$T = \left(\bigcup_{j \in J_3} T_{a(j)}\right) \cup \left(\bigcup_{j \in J_2} T_{a(j)}'\right)$$

and

$$R = \left(\bigcup_{j \in J_3} R_{a(j)}\right) \cup \left(\bigcup_{j \in J_2} R_{a(j)}'\right)$$

which means T consists of all triples or pairs of stream segments that constrain each other at junctions, and R is the corresponding segment–label constraint relation.

Now the labeling problem of assigning {upstream, downstream} to all stream segments can be described by a compatibility model (S, L, T, R) which is a particular instance of the general consistent labeling problem. Because there are many spatial inference problems which are instances of the consistent labeling problem, the form of the general consistent labeling problem as given by Ullman, Haralick, and Shapiro [19] is reviewed in the next section.

Consistent Labeling

Let U be a set of objects called units, and L be a set of possible labels for those units. Let $T \subseteq \{f | f \subseteq U\}$ be the collection of those subsets of units from U that mutually constrain one another. That is, if $f = \{u_1, u_2, \cdots, u_k\}$ is an element of T, then not all possible labelings of $u_1, \cdots, u_k$ are legal labelings. Thus there is at least one label assignment $\ell_1, \ell_2, \cdots, \ell_k$ so that u_1 having label ℓ_1, u_2 having label $\ell_2, \cdots, u_k$ having label ℓ_k is a forbidden labeling. T is called the unit constraint set. Finally, let $R \subseteq \{g | g \subseteq UxL$, g single-valued, and $\text{Dom}(g)$ in $T\}$ be the set of unit-label mappings in which constrained subsets of units are mapped to their allowable subsets of labels. If $g = \{(u_1, \ell_1), (u_2, \ell_2), \cdots, (u_k, \ell_k)\}$ is an element of R, then $u_1, u_2, \cdots, u_k$ are distinct units, $\{u_1, u_2, \cdots, u_k\}$ is an element of T meaning $u_1, u_2, \cdots, u_k$ mutually constrain one another, and u_1 having label ℓ_1, u_2 having label $\ell_2, \cdots,$ and u_k having label ℓ_k are all simultaneously allowed.

In the consistent labeling problem, one is looking for functions that assign a label in L to each unit in U and satisfy the constraints imposed by T and R. That is, a consistent labeling is one which when restricted to any unit constraint subset in T yields a mapping R. In order to state this more precisely, the restriction of mapping is first defined. Let h: $U \rightarrow L$ be a function that maps each unit in U to a label in L. Let $f \subseteq U$ be a subset of the units. The restriction $h|f$ (read h restricted by f) is defined by $h|f = \{(u, \ell) \in h | u \in f\}$. With this notation, a consistent labeling is defined as follows.

A function h: $U \rightarrow L$ is a consistent labeling if and only if for every $f \in T$, $h|f$ is an element of R.

An example is given below. Suppose the inputs to the problem are as follows:

$$\begin{aligned}
U &= \{1, 2, 3, 4, 5\} \\
L &= \{a, b, c\} \\
T &= \{\{1\}, && \text{unary constraint} \\
&\quad \{1, 2\}, && \text{binary constraints} \\
&\quad \{2, 5\}, \\
&\quad \{1, 3, 4\}\} && \text{ternary constraint} \\
R &= \{\{(1, a)\}, \{(1, b)\}, && \text{unary constraint} \\
&\quad \{(1, a), (2, a)\}, \\
&\quad \{(1, a), (2, b)\}, \\
&\quad \{(1, b), (2, b)\}, && \text{binary constraints} \\
&\quad \{(2, a), (5, a)\}, \\
&\quad \{(2, b), (5, c)\}, \\
&\quad \{(1, a), (3, a), (4, c)\}, && \text{ternary constraints} \\
&\quad \{(1, b), (3, a), (4, a)\}\}.
\end{aligned}$$

Then $h = \{(1, a)\ (2, a)\ (3, a)\ (4, c)\ (5, a)\}$ is a consistent labeling. To see this note that $h|\{1\} = \{(1, a)\}$, $h|\{1, 2\} = \{(1, a), (2, a)\}$, $h|\{2, 5\} = \{(2, a), (5, a)\}$, and $h|\{1, 3, 4\} = \{(1, a), (3, a), (4, c)\}$ are all elements of R.

If having $\ell_1, \cdots, \ell_k$ applied to $u_1, \cdots, u_k$ when $\{(u_1, \ell_1), \cdots, (u_k, \ell_k)\}$ is not in R is allowed with a penalty, the process is called inexact consistent labeling [16]. In order to include these mappings, an error weighting function Ew is defined as Ew: $G \rightarrow [0, 1]$, where $G \subseteq \{g | g \subseteq UxL$, g single-valued and $\text{Dom}(g) \in T\}$. $Ew(\{(u_1, \ell_1), (u_2, \ell_2), \cdots, (u_k, \ell_k)\})$ is the error which occurs when labels $\ell_1, \ell_2, \cdots, \ell_k$ are applied to $u_1, u_2, \cdots, u_k$.

If $\{(u_1, \ell_1), \cdots, (u_k, \ell_k)\}$ is in R, Ew is zero; otherwise, Ew is a constant ec and usually is defined as the reciprocal of the square of the size of U. The mapping h: $U \rightarrow L$ is an inexact consistent labeling if for all f in T, the sum of $Ew(h|f)$ is within some upper bound, usually 1.

Relational Reasoning Model and Flow Direction of Streams

In the last section, the very general consistent labeling model was introduced and the unit–label pairs in the elements of R were just assumed to be there. However, if one goes back to the flow direction problem and looks at Table 1, it is clear that one cannot talk about unit–label pairs without looking at the property values, such as angles and lengths, of these units. In the following, based on the very general consistent labeling model, a relational reasoning model is defined to explicitly include these properties. However, these properties are related only to creation of elements in the set R; the basic tree-searching technique is just the same for both the general consistent labeling model and the relational reasoning model. We first show that the relational reasoning model is applicable to the flow direction problem, and then discuss the tree searching strategies designed by Shapiro and Haralick [16].

In relational reasoning problems, many spectral and geometrical properties can be computed for locally detected units. Some frequently used properties are average gray-level, size, and shape descriptors. For each unit, a list of property values can be computed. Considering all the units, these lists form an array which can be named P. Thus for a unit u, $P[u]$ gives the list of property values for u. For stream junctions, the line length of one segment and the clockwise angle from one segment to the next one can be detected so the $P[s] =$ (length, angle) for a unit s. For example, in Fig. 14, $P[s_1] = (10, 45)$, $P[s_2] = (10, 135)$, and $P[s_3] = (15, 180)$.

For each junction pattern in Table 1, the angles must be within certain ranges. With respect to pattern number 5, $P[s_1]$ must be in the property range $([1, ub]), [0, 89])$, $P[s_2]$ must be in the range $([1, ub], [91, 179])$, and $P[s_3]$ must be in the range $([1, ub], [-180, 180])$ for some upper bound ub on the line length.

However, simply specifying a range for each unit is not enough. Sometimes one needs to compare the property values for different units. One example for the stream junctions is the pattern "$L(s_3) \geqslant \max(L(s_1), L(s_2))$." To handle this type of constraint, a relation $r(P[u_1], \cdots, P[u_k])$ must be defined on the property lists of the related units.

Now the relational reasoning model is a six-tuple (U, P, L, T, R, Ew). U, L, T, Ew have the same meaning as before; however, the elements in R now have the form $\{(u_1, p_1, \ell_1), \cdots, (u_k, p_k, \ell_k); r(P[u_1], P[u_2], \cdots, P[u_k])\}$ where p_i is the list of the required ranges of property value for all the properties in P for unit u_i, $i = 1$ to k. If the property values of u_i are within the ranges specified by p_i for $i = 1$ to k, $\{u_1, \cdots, u_k\}$ is contained in T, and relation r is satisfied, then it is legal to assign lable ℓ_1 to $u_1, \cdots, \ell_k$ to u_k at the same time.

The relational reasoning model (U, P, L, T, R, Ew) can be applied to deduce the flow directions of visible rivers. U contains the units of visible rivers. P contains all the properties detectable from the stream segments. The most important properties are the length of a segment and the orientation of the segment at both ends because they are used in Table 1. L is $\{\text{upstream} = 1, \text{downstream} = 2\}$. T contains the junction relations. R contains the relations of legal flow directions defined in Table 1. For $\{u_1, \cdots, u_k\}$ in T, if $\{(u_1 p_{1j}, \ell_1), \cdots, (u_k, p_k, \ell_k)\}$ is in R, the error function $Ew(\{(u_1, p_1, \ell_1), \cdots, (u_k, p_k, \ell_k)\})$ is defined to be zero; otherwise, it is ec, the reciprocal of the square of the total number of stream segments.

To find the best possible labeling, four different tree-searching strategies were described for solving the inexact consistent labeling problem [16]. Experiments were done to evaluate their performance. Forward checking was found the most efficient one. In the following, the idea of the forward checking strategy is described first in English and then followed by mathematical equations.

A tree search finds a label for each unit. Each node of the tree represents a possible assignment of a label ℓ to a unit u. Associated with such a node is 1) the past error, 2) the error of this instantiation, and 3) the future error. Past error consists of the error of the partial mapping defined by the ancestors of this node in the tree. This error is the sum of $Ew(h|f)$ for all $f \in T$. Error of instantiation is the error induced by the assignment of label ℓ to unit u.

In a simple backtracking tree search, the error of instantiation is computed at the time ℓ is assigned to u. In a tree search with forward checking, an error table keeps track of how much error the assignment of any label to an uninstantiated unit will generate. This is accomplished by constructing an updated table each time an assignment of ℓ' to u' is made and propagating forward to each pair of as-yet-unassigned unit u'' and possible label ℓ'' in that table the error that would be caused by a simultaneous assignment of u' to ℓ' and u'' to ℓ''.

At any node of the tree, each as-yet-uninstantiated unit has a label in the error table with minimal propagated error. The sum of the minimum error for each such unit is the future error.

If at any node of the tree, the sum of the past error, error of instantiation, and future error is greater than the allowable threshold, then the current assignment at this node is not made and the next label assignment is tried. If there is no such next label assignment possibility left, then backtracking occurs. Otherwise, the error of this assignment is propagated forward and the tree search continues. Details of these are given below.

The inexact consistent labeling problem can be solved by a brute-force backtracking tree search. Before the bottom of the tree is reached, only some of the units are labeled, and thus only the error incurred against all units which have already been assigned labels can be calculated. Such a labeling is called a partial labeling; the labeled units are called past units, and the set of all past units is called U_p. Similarly, the units which have not been labeled are called future units, and the set of all future units is called U_f. Also let $T1$ be the set of all sets composed of units which have already been assigned labels, i.e., $T1 = \{\{u_1, u_2, \cdots, u_k\} | u_1, u_2, \cdots, u_k \in Up \text{ and } \{u_1, u_2, \cdots, u_k\} \in T\}$. Thus the error for past units, ep, incurred in backtracking is

$$ep(Up, h) = \sum_{\{u_1, \cdots, u_k\} \in T1} Ew(\{(u_1, h(u_1)), \cdots, (u_k, h(u_k))\}) \quad (13)$$

for a partial labeling h. If the error sum exceeds an error bound eb, the tree search must either try the next label for the current unit or if there is no next label, it must backtrack.

As a simple example, let $U = \{1, 2, 3, 4, 5\}$, $L = \{a, b, c\}$, $T = \{\{1\}, \{1, 2\}, \{1, 4\}\}$, $R = \{\{(1, a)\}, \{(1, a), (2, b)\}, \{(1, b), (2, b)\}, \{(1, b), (4, b)\}\}$, error constant $ec = 1/6$, and error bound $eb = 0.2$. In the tree search, label a is assigned to 1 first. Thus $Up = \{1\}$, $T1 = \{\{1\}\}$, $ep(Up, h) = 0$ because $\{(1, a)\}$ is in R.

Next, label a is assigned to 2 because backtracking is depth-first. Now $Up = \{1, 2\}$, $T1 = \{\{1\}, \{1, 2\}\}$, $ep(Up, h) = 1/6$ because $h = \{(1, a), (2, a)\}$ is not in R. Since ep is smaller than $eb = 0.2$, one can continue and assign label a to unit 3. As $\{1, 3\}$, $\{2, 3\}$ are not in T, $T1$ is not changed and $ep(Up, h)$ is not changed.

Next, label a is assigned to unit 4 which will cause $Up = \{1, 2, 3, 4\}$, $T1 = \{\{1\}, \{1, 2\}, \{1, 4\}\}$, $ep(Up, h) = 1/6 + 1/6 = 1/3$ because $\{(1, a), (4, a)\}$ is not in R. At this point, $ep = 1/3$ which is larger than $eb = 0.2$, and one cannot continue with unit 5. Instead, next label b is assigned to unit 4. This is the trace for backtracking.

A technique called backtracking with forward checking can improve the speed of tree search. For the previous example, 3 units were assigned labels before a cutoff happened. Actually, by looking at sets T, R and doing some calculations described below, a decision about cutoff can be made even after the first unit is assigned a label. Thus the searching is more efficient. This technique is similar to the branch-and-bound technique except that a fixed bound value is used.

The speed of tree search can be improved if one also considers the minimum error that the current labeling must incur against future units which have not been assigned labels. Thus the set in T containing only one future unit is of interest; for a future unit u and label ℓ, define

$$T(u,i;Up) = \{\{u_1,\cdots,u_k\} \text{ in } T | u_i = u \text{ and } n \neq i \text{ implies } u_n \text{ in } Up\}.$$

For example, when $Up = \{1\}$, $h = \{(1,a)\}$, $T(2,2;\{1\}) = \{\{1,2\}\}$, $T(4,2;\{1\}) = \{\{1,4\}\}$.

Using labeling h on all units except u and assigning label ℓ to u, the error (epf, error for past and future units) is

$$\begin{aligned} &\mathrm{epf}(u,\ell;Up,h) \\ &= \sum_{i=1}^{k} \sum_{\substack{\{u_1,\cdots,u_k\} \\ \text{in } T(u,i;Up)}} Ew(\{(u_1,h(u_1)),\cdots,(u_{i-1},h(u_{i-1})), \\ &\qquad (u,\ell),(u_{i+1},h(u_{i+1})),\cdots,(u_k,h(u_k))\}). \end{aligned} \tag{14}$$

In the continuing example, if $u = 2$, $\ell = a$, then $\mathrm{epf}(2,a;\{1\};\{(1,a)\}) = Ew(\{(1,a),(2,a)\}) = 1/6$ because $\{(1,a),(2,a)\}$ is not in R.

To be complete, one should also consider the smallest error of the units in the nodes with higher level numbers in the tree created by backtracking or the units other than u in Uf. It is

$$\sum_{\substack{v \in Uf \\ v \neq u}} \min_{m \in L} \mathrm{epf}(v,m;Up,h). \tag{15}$$

For the continuing example, when $v = 3$

$$\min \mathrm{epf}(3,m;\{1\}\{(1,a)\}) = 0, \qquad m \in L$$

because $T(3,i;\{1\})$ is always empty.

When $v = 4$;

$$\mathrm{epf}(4,a;\{1\}\{(1,a)\}) = 1/6$$
$$\mathrm{epf}(4,b;\{1\}\{(1,a)\}) = 1/6$$
$$\mathrm{epf}(4,c;\{1\}\{(1,a)\}) = 1/6$$

and

$$\min \mathrm{epf}(4,m;\{1\},\{(1,a)\}) = 1/6, \qquad m \in L.$$

When $v = 5$,

$$\min \mathrm{epf}(5,m;\{1\},\{(1,a)\}) = 0, \qquad m \in L$$

for the same reason as when $v = 3$. Now the sum in (15) is $0 + 1/6 + 0 = 1/6$.

For current labeling h, if the summation of (13)–(15) exceeds an error bound for any label ℓ for the current unit u, then one needs either try the next label for the current unit or backtrack. This is called backtracking tree search with forward checking. From the above calculations, for $u = 2$, $\ell = a$, $0 + 1/6 + 1/6 = 1/3 > 0.2$, and one needs to try the next label b for current unit 2. Thus it is clear that only one node is generated in the searching tree as opposed to three nodes in the case of backtracking.

Implementing the model (U,P,L,T,R,Ew) by using the algorithm of forward checking, the flow directions for the test area have been deduced. They are labeled as in Fig. 15. The valley segments that are used to help make the decision are also shown in Fig. 15. The interpreted flow directions are correct with respect to actual flow directions.

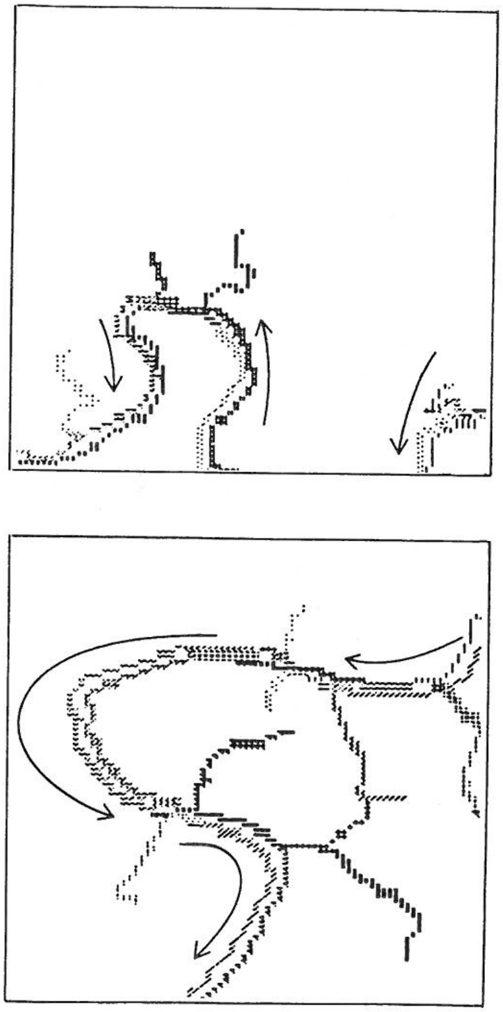


Fig. 15. Interpreted flow directions within two segments of the test image. Separate symbols designate individual segments defined in the analysis. Arrows indicate interpreted flow directions, which agree with correct flow directions from ground data.

Discussion

Here we have described the derivation of estimates of relative elevation from a single multispectral view of a region of uneven topography. The procedure applies a few simple rules and assumptions concerning the behavior of solar radiation as it interacts with the earth's atmosphere to illuminate the ground surface. Application of these rules permits us to separate individual contributions of the atmosphere, direct illumination, indirect illumination, and spectral reflectance of varied materials at the earth's surface. Separation of these components enables isolation of those variations in brightness that are due solely to variations in topographic relief. It is this topographic image that is then used in subsequent steps of the analysis. From the pattern of highlighted and shadowed areas, the positions of ridgelines and valleys can be estimated. From these, we apply knowledge of the general configuration of the earth's topography—the spatial arrangements of peaks, ridgelines, and stream networks. Realistic modeling of local topography depends upon approximations of a local slope within the imaged region derived either from direct observation, or from inferences based upon general knowledge of regional geology and geomorphology.

Finally, our topographic image can be examined in more detail. The network of valleys and streams must follow certain rules concerning the directions of flow and the nature of valley intersections. By application of the consistent labeling process we determine correct direction of stream flow, thereby refining our estimate of topography.

We defined a set of constraint rules for stream junctions, then applied them to the units detected from the topographic analysis. Even though the numbers of units in the test area are not large, the mathematical expression of the spatial reasoning model is precise and can be useful in more complex applications.

References

[1] T. T. Alfoldi and J. C. Munday, Jr., "Water quality analysis by digital chromaticity mapping of Landsat data," *Can. J. Remote Sensing*, vol. 4, no. 2, pp. 108–126, 1978.

[2] J. Bryant, "On the clustering of multidimensional pictorial data," *Pattern Recogn.*, vol. II, pp. 115–125, 1979.

[3] J. B. Campbell, R. W. Ehrich, D. Elliott, R. M. Haralick, and S. Wang, "Spatial reasoning in remotely sensed data," in *Proc. 15th Int. Symp. on Remote Sensing of Environment*, vol. I, pp. 223–235, 1981.

[4] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* New York: Wiley, 1973.

[5] P. T. Eliason, L. A. Soderblom, and P. S. Chavez, "Extraction of topographic and spectral albedo information from multispectral images," *Photogramm. Eng. Remote Sensing*, vol. 48, pp. 1571–1579, 1981.

[6] M. Goldberg and S. Shlien, "A clustering scheme for multispectral images," *IEEE Trans. Systems. Man, Cybern.*, vol. SMC-8, no. 2, 1978.

[7] R. M. Haralick and L. G. Shapiro, "The consistent labeling problem: Part 1," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-1, no. 2, Apr. 1979.

[8] R. M. Haralick and L. G. Shapiro, "The consistent labeling problem: Part 2," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, no. 3, May 1980.

[9] R. M. Haralick and S. Wang, "Relative elevation determination from Landsat imagery," in *Proc. COMPCON* (Washington, DC, 1983).

[10] R. M. Haralick, S. Wang, and D. B. Elliott, "Spatial reasoning to determine stream network from Landsat imagery," presented at the 6th Int. Conf. on Pattern Recognition, Munich, Germany, 1982.

[11] A. W. Kuchler, *Potential Natural Vegetation of the Conterminous United States. Amer. Geographical Soc. Special Publication* 36, 1964.

[12] J. G. Moik, *Digital Processing of Remotely Sensed Images.* NASA Special Publication 431, 1980.

[13] N. C. Mulder, "Methodology of color coding MSS and other data," presented at Inter-Congress Symp., Commission III (Mathematical Models, Accuracy Aspects, and Quality Control), Int. Soc. of Photogrammetry and Remote Sensing, Otaniemi, Finland, 1982.

[14] National Cartographic Information Center, *Digital Terrain Tapes, User Guide*, 2nd. ed., 1980.

[15] A. G Roy, "Optimal angular geometry models of river branching," *Geographical Anal.*, vol. 15, no. 2., pp. 87–96, Apr. 1983.

[16] L. G. Shapiro and R. M. Haralick, "Structural description and inexact matching," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-3, no. 5, Sept. 1981.

[17] P. Switzer, W. S. Kowalik, and R. J. P. Lyon, "Estimation of atmospheric path-radiance by the covariance matrix method," *Photogramm. Eng. Remote Sensing*, vol. 48, pp. 1469–1476, 1981.

[18] W. D. Thornbury, *Regional Geomorphology of the United States.* New York: Wiley, 1967.

[19] I. R. Ullmann, R. M. Haralick, and L. G. Shapiro, "Computer architecture for solving consistent labeling problem," in *Proc. SSST*, 1982.

[20] R. K. Vincent, "Ratio maps of iron ore deposits, Atlantic City District, Wyoming," in *Proc. Symp. on Significant Results Obtained from the ERTS-1*, pp. 379–386, 1973.

[21] S. Wang, D. B. Elliott, J. B. Campbell, R. W. Ehrich, and R. M. Haralick, "Spatial reasoning in remotely sensed data," *IEEE Trans. Geosci. Remote Sensing*, vol. GE-21, pp. 94–101, 1983.

[22] S. Wang, R. M. Haralick, and J. B. Campbell, "Relative elevation determination from Landsat imagery," Spatial Data Analysis Lab. Tech. Rep., Virginia Polytech. Inst. and State Univ., Blacksburg, 1983.